

TP13 – Techniques de photomontage

Exercice 1 : photomontage par collage

Le *photomontage par collage* consiste à remplacer une zone d'une image « cible », notée c , par des données issues d'une image « source », notée s . Le script `exercice_0` demande à l'utilisateur de sélectionner un polygone p dans s , puis de sélectionner un rectangle r dans c (en cliquant sur deux sommets opposés). La fonction `imresize` de Matlab permet de définir une transformation affine $t : s \rightarrow c$, telle que $t(e) = r$, où e est le rectangle englobant de p . Le résultat est une image u telle que $u(x, y) = c(x, y)$ partout, sauf pour les points $(x, y) \in t(p)$, auquel cas $u(x, y) = s(t^{-1}(x, y))$. Ce « collage » doit être effectué canal par canal. Vous constatez que le résultat n'est pas réaliste, ce qui n'est guère surprenant avec un algorithme aussi naïf.

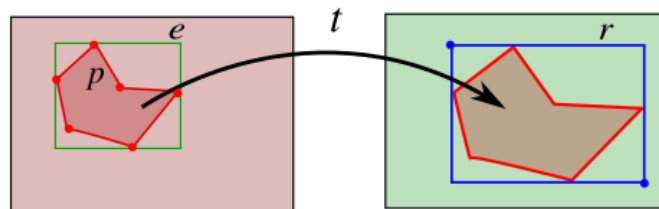


FIGURE 1 – À gauche : sélection d'un polygone p dans l'image source s . À droite : sélection d'un rectangle r dans l'image cible c . Les points cliqués par l'utilisateur sont indiqués en rouge dans s , en bleu dans c .

Au lieu d'insérer dans c des données issues de s , on peut résoudre l'équation $\nabla u = \mathbf{g}$ sur r , où le champ vectoriel \mathbf{g} vaut ∇c sur $r \setminus t(p)$ et ∇s sur $t(p)$. Si l'on impose $u = c$ sur le bord de r , ce problème n'a en général aucune solution exacte. On cherche donc à le résoudre de manière approchée, au sens des moindres carrés :

$$\min_{u: \mathbb{R}^2 \rightarrow \mathbb{R}} \iint_{(x,y) \in r} |\nabla u(x,y) - \mathbf{g}(x,y)|^2 dx dy \quad (1)$$

L'équation d'Euler-Lagrange associée à ce problème est l'équation de Poisson :

$$\Delta u(x,y) = \nabla \cdot \mathbf{g}(x,y), \quad (2)$$

Cette équation doit être résolue canal par canal. La discrétisation de (2) par différences finies s'écrit :

$$\mathbf{A} \mathbf{u}_k = \mathbf{b}_k, \quad k \in \{R, V, B\} \quad (3)$$

Dans (3), la matrice \mathbf{A} est une version discrète de l'opérateur laplacien, de taille $N \times N$, où N est le nombre de pixels contenus dans r , et les vecteurs \mathbf{u}_k et \mathbf{b}_k concatènent, respectivement, les valeurs de u (inconnues) et les valeurs de $\nabla \cdot \mathbf{g}$ (connues) calculées pour l'ensemble des pixels de r , dans le canal k .

Remarques :

- Dorénavant, contrairement à l'algorithme naïf ci-dessus, qui ne modifie que les points $(x, y) \in t(p)$, tous les points de $(x, y) \in r$ sont modifiés. L'intérêt d'utiliser un domaine r de forme rectangulaire est que cela donne à la matrice \mathbf{A} une structure très régulière (cf. TP11).
- Il ne faut surtout pas remplacer le laplacien de c par le laplacien de s à l'intérieur de $t(p)$: il faut remplacer le gradient de c par le gradient de s , puis calculer la divergence $\nabla \cdot \mathbf{g}$ du champ vectoriel ainsi formé.

Conseils pour l'écriture de la fonction `collage`, qui doit appliquer cette nouvelle procédure de photomontage :

- Convertissez les paramètres d'entrée `r` et `s` au format double.
- Pour affecter à `bord_r` la liste des indices des pixels du bord de `r`, il est commode de construire une matrice de 1 de mêmes dimensions que `r`, et de mettre l'intérieur de cette matrice à 0.
- Pour imposer la condition $u = c$ sur le bord de `r`, modifiez les lignes de la matrice `A` et les lignes du vecteur `bk` correspondants, en écrivant (`nb_bord_r` et `nb_r` désignent des nombres de pixels) :

$$A(\text{bord_r}, :) = \text{sparse}(1:\text{nb_bord_r}, \text{bord_r}, \text{ones}(\text{nb_bord_r}, 1), \text{nb_bord_r}, \text{nb_r});$$

$$\dots$$

$$b(\text{bord_r}) = u_k(\text{bord_r});$$

Faites une copie de `exercice_0`, de nom `exercice_1`, que vous modifierez de manière à remplacer l'appel à la fonction `collage_naif` par `collage`. Ce script vous permettra d'incruster la randonneuse du TP12 dans un tableau de Van Gogh (cf. figure 2), ou de réaliser d'autres photomontages avec la paire d'images de votre choix.



FIGURE 2 – À gauche : image source s . Au centre : image cible c . À droite : résultat u du photomontage.

Exercice 2 : décoloration partielle d'une image

D'autres techniques de photomontage sont décrites dans un article datant de 2003, dont ce TP s'inspire en grande partie : <https://www.cs.jhu.edu/~misha/Fall107/Papers/Perez03.pdf>

La fonction `rgb2gray` de Matlab permet de transformer une image couleur en image en niveaux de gris. Cette transformation est également réalisable en utilisant le format LAB, ayant comme premier canal la luminance (cf. TP12). Les fonctions `rgb2lab` et `lab2rgb` de Matlab permettent de passer du format RGB au format LAB. L'exemple de la figure 3 illustre la *décoloration partielle d'une image*, où l'image originale constitue la source s , et où le canal de luminance de cette même image constitue la cible c . Ce qui est remarquable, avec cette autre technique de photomontage, c'est qu'elle ne nécessite pas de segmentation préalable de la rose. Il suffit que le polygone p , qui est visible sur l'image de gauche de la figure 3, détoure très grossièrement la rose sur l'image originale. Attention : n'allez pas croire que cette technique effectue, comme par magie, la segmentation de la rose !

Faites une copie du script `exercice_1`, de nom `exercice_2`, que vous modifierez de manière à initialiser la source s et la cible c par l'image `rose.jpg` convertie au format LAB, puis à annuler les deuxième et troisième canaux de l'image c . La fonction `collage` reste inchangée. Là encore, une fois mis au point, testez le script `exercice_2` sur d'autres images que `rose.jpg`.

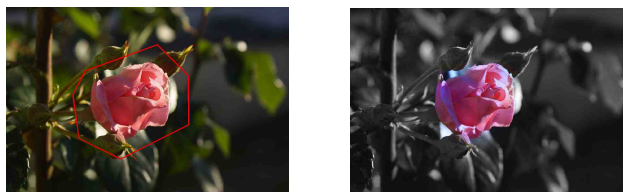


FIGURE 3 – À gauche : image originale. À droite : décoloration partielle de l'image sans segmentation préalable.